Assignment 1:   
Knowledge-aware Timeline Summarization

Due: 03/11

# Goal

In this assignment, you will get familiar with timeline summarization task, including the input and output, and evaluation metrics. You will implement a timeline summarization system which can incorporate the structured knowledge, such as the events and their arguments.

# Data

1. Timeline dataset: Please download the dataset in <https://uofi.box.com/s/wqap6k1k2jy86b9ytvsenl3qdmmw2dl1>, with the dataset description in <https://uofi.box.com/s/xr6zg07d8txbbht0jk5v0v2512yxr858>.
2. Information Extraction (IE) results (event graphs) of input documents, including entities and events: <https://uofi.box.com/s/91xdhxd0zgg855jji4yj2nxh0vwx4u65>. The format is in Cold Start format (<https://docs.google.com/document/d/1DS2TX2syeJ8Xzy0fbhZe0YaqsU5NCqOTedtXZbLczuM/edit?usp=sharing>).

If you have any questions about data, please contact the TA ([manling2@illinois.edu](mailto:manling2@illinois.edu)). Thanks.

# 3 Baseline

1. Extractive summarization baseline:

<https://github.com/smartschat/tilse/blob/master/SUMMARIZATION.md>

1. Abstractive summarization baseline:

<https://github.com/julmaxi/Abstractive-Timeline-Summarization>

# 4 Evaluation Code

Please find the evaluation code in <https://github.com/smartschat/tilse>.

The example code is as follows.

|  |
| --- |
| from tilse.data import timelines  from tilse.evaluation import rouge  evaluator = rouge.TimelineRougeEvaluator(measures=["rouge\_1"])  predicted\_timeline = timelines.Timeline({  datetime.date(2010, 1, 1): ["Just a test .", "Another sentence ."],  datetime.date(2010, 1, 3): ["Some more content .", "Even more !"]  })  groundtruth = timelines.GroundTruth(  [  timelines.Timeline({  datetime.date(2010, 1, 2): ["Just a test ."],  datetime.date(2010, 1, 4): ["This one does not match ."]  }),  timelines.Timeline({  datetime.date(2010, 1, 5): ["Another timeline !"],  })  ]  )  pp = pprint.PrettyPrinter(indent=4)  print("concat")  pp.pprint(evaluator.evaluate\_concat(predicted\_timeline, groundtruth))  print("")  print("align, date-content costs")  pp.pprint(evaluator.evaluate\_align\_date\_content\_costs(predicted\_timeline, groundtruth)) |

The metrics include:

* + *Date F1 :*to evaluate the key date selection using, and the content generation using ROUGE scores
  + *concat* *F1* :   
    to compute ROUGE by concatenating the summaries of all selected dates
  + *agree* *F1* :   
    to compute ROUGE only between the summaries which have the same dates
  + *align* *F1* :   
    to first align summaries in the output with those in the reference based on similarity and the distance between their dates, then compute the ROUGE score between aligned summaries.

# 5 Submission

Please name your submission as `netid\_assignment1.zip`, including a report named as `netid\_report.pdf` and the code `netid\_code`. The code should include a README.md with environment and running instructions.

The submission is done through Blackboard Compass2g (<https://compass2g.illinois.edu/webapps/blackboard/content/listContentEditable.jsp?content_id=_5671021_1&course_id=_62362_1> )

# 6 Tasks

1. **(3pt)** Apply any baseline model to generate timelines.
2. **(3pt)** Evaluate and report the evaluation scores of the selected baseline.
3. **(5pt)** Use Information Extraction (IE) results to make the model entity-aware, relation-aware, event-aware, etc. The extraction output is provided and detailed in the data section (Section 2). The grades will be assigned based on the rank of your improved system among all systems from the class.
4. **(2pt)** Writeclear and informative written report about your methods, results and findings.
5. **(2pt)** Writeclear README of the submitted code, including environment and running instructions.

# Bonus (up to 5pts)

1. Leveraging external resources, such as Wikipedia, or GPT-3.
2. Better strategy to decide the compression rate/generation length.
3. Better evaluation metrics; report results using the new metrics.
4. Suggestions about large-scale timeline data collection.